Review Sheet for Midterm 1 Selected Solutions APPM/MATH 4650 Fall '20 Numerical Analysis

Instructor: Prof. Becker
solutions version 9/23/2020

Chapter 1: basics

See the extra nodes on machine epsilon attached to this PDF.

1. Why do we use floating point numbers? What are the alternatives? What are the advantages/disadvantages?

   Solution:

   They let us efficiently store very large and small numbers, and have fairly good accuracy. Alternatives would be as integers or fractions (but these can then take up more space as we do computation), or fixed point (which is not good at dealing with large scales). Basically, there's no good alternative, though these alternatives are used in specialized settings.

2. When is a good time to use absolute accuracy/error (and what are the definitions)? When is a good time to use relative accuracy/error?

   Solution:

   Sometimes a problem specifies a clear absolute accuracy, like designing a machine part to be within a 1mm tolerance. Othertimes we like relative accuracy because it corresponds to digits of precision, and floating point numbers have a fixed relative accuracy (about 10^{-16}) while their absolute accuracy depends on their magnitude.

   If y is the true value and $\hat{y}$ is our approximation (e.g., maybe $y = f(x)$, and $\hat{y}$ is the result of our algorithm), then absolute accuracy/error is $|\hat{y} - y|$ (of course, order doesn't matter with absolute value, so we could just as well write $|y - \hat{y}|$.

   Relative accuracy/error is $|\hat{y} - y|/|y|$.

3. How does "number of correct digits" relate to relative accuracy? To absolute accuracy?

   Solution:

   The number of correct digits is $-\log_{10}$(relative accuracy). If relative accuracy is 10^{-4}, then we have 4 correct digits. For relative accuracy of 5×10^{-4}, you could say it has either 3 or 4 digits; we are usually not that precise when we discuss "digits".

   Number of correct digits (for floating point numbers) has no easy connection to absolute accuracy, since it would depend the size of the number.

4. On a computer, is $a + (b + c) = (a + b) + c$ always true? Can you think of examples?

   Solution:

   No, e.g., $a = 1$ and $c = -1$ and $b = \epsilon_{\text{mach}}/2$.

5. What's the difference between *conditioning* and *stability*?

Solution:

Conditioning is about a math problem ("find the roots of f"), whereas stability is about an algorithm to solve the problem. If a problem has condition number $\kappa = 10^5$, then we expect to multiple ϵ_{mach} by κ to get the final relative accuracy; e.g., we lose $\log_{10}(\kappa)$ digits of accuracy no matter how clever our algorithm is.

If the condition number of any piece of the algorithm is greater than the condition number of the math problem, then we've lost extra accuracy that we didn't have to lose.

6. Is it more accurate to evaluate a function f at x if $|f'(x)|$ is large or small?

Solution:

When $|f'(x)|$ is small. E.g., $f(x) = c$ a constant function, then we can evaluate this very stably for any input.

7. Is it more accurate to find a root p of the function f if $|f'(p)|$ is large or small?

Solution:

When $|f'(p)|$ is large

8. What is the relative condition number? (formula and meaning)

Solution:

Formula is $\kappa_f^{\text{rel}}(x) = |\frac{x}{f(x)} f'(x)|$, and it means that the relative error will be about $\epsilon_{\text{mach}} \cdot \kappa_f^{\text{rel}}(x)$.

9. How does the relative condition number relate to relative error and the number of correct digits?

Solution:

Relative error is about $\epsilon_{\text{mach}} \cdot \kappa_f^{\text{rel}}(x)$. If the condition number is $\kappa_f^{\text{rel}} = 10^d$, then we expect to *lose* about d digits, relative to the number of accurate digits we started with.

10. What is the relative condition number of $f(x) = x^2 + 10^{10}$ at $x = 1$? (an approximate value is OK). Can you interpret this? Can we take advantage of this in double precision floating point?

Solution:

Writing

$$\kappa_f^{\text{rel}}(x) = \frac{x}{f(x)} f'(x) = \frac{x}{x^2 + 10^{10}} 2x \approx 10^{-10} \text{ if } x = 1$$

This seems to imply that we will *gain* digits. This makes sense in this case: if we know 10^{10} precisely, we're now forming a number that has about 26 accurate digits. Unfortunately, usually we cannot take advantage of this, because in double precision floating point we can't store that many digits. It could help if we'd lost digits already.

Think of it like this: if at some point in the calculation we use the function $f(x) = 3$ (or any constant function), then even if we've lost accuracy previously, we regain that accuracy and "wipe the slate clean" because regardless of how in accurate x is, the output of f is correct. You can check that since $f'(x) = 0$, we have $\kappa_f^{\text{rel}}(x) = 0 = 10^{-\infty}$, i.e., we've gained an infinite amount of digits (assuming that the constant 3 is known to perfect accuracy).

11. Consider the approximation $\cos(x) \approx 1 - x^2/2$. What is the *absolute* error of this approximation on the interval $[-.1, .1]$? [we are asking about mathematical error, now stability issues of an *algorithm*].

Solution:

This approximation is from the Taylor series of cos expanded about $x = 0$. By the Taylor remainder theorem, $\cos(x) = \cos(0) - x\sin(0) - x^2/2\cos(0) + x^3/3!\sin(\xi)$ for some $\xi \in [0, x]$. Thus the error is $|x^3/3!\sin(\xi)| \leq |x^3/6| \leq .1^3/6 = 1.6\overline{6} \cdot 10^{-4}$.

12. True/false? If f is continuous then $\forall x > 0$, $\exists \xi \in [0, x]$ such that $\int_0^x f(y)\, dy = x f(\xi)$?

Solution:

This is the integral form of the MVT (e.g.., using the FTC). The answer is True.

13. What does it mean to say $f(n) = O(n^2)$ as $n \to \infty$? Is $n = O(n^2)$? Is $n^3 = O(n^2)$?

Solution:

It means $\limsup_{n\to\infty} \frac{f(n)}{n^2} < \infty$ (and often the limit exists, so we can replace the lim sup with just a lim). In other words, there is a constant c and an integer N such that if $n \geq N$ then $\frac{f(n)}{n^2} \leq c$.

It is true $n = O(n^2)$, but false that $n^3 = O(n^2)$.

14. What does it mean to say $f(n) = o(n^2)$ as $n \to \infty$?

Solution:

$\limsup_{n\to\infty} \frac{f(n)}{n^2} = 0$, or that for *all* constants c, there is an integer N (which may depend on c) such that if $n \geq N$ then $\frac{f(n)}{n^2} \leq c$.

Thus $n^2 \neq o(n^2)$ but $n = o(n^2)$.

15. What does it mean to say $f(n) = \Theta(n^2)$ as $n \to \infty$?

Solution:

It means $f = O(n^2)$ and $n^2 = O(f(n))$.

16. What does it mean to say $f(h) = O(h^2)$ as $h \to 0$? Is $h = O(h^2)$? Is $h^3 = O(h^2)$?

Solution:

It means $\limsup_{h\to 0} \frac{f(h)}{h^2} < \infty$ (and often the limit exists, so we can replace the lim sup with just a lim). In other words, there is a constant c and a value δ such that if $|h| \leq \delta$ then $\frac{f(h)}{h^2} \leq c$.

Then $h \neq O(h^2)$ but $h^3 = O(h^2)$.

17. What is quadratic convergence?

Solution:

If $e_n \to 0$ we say it has quadratic convergence if

$$\lim_{n\to\infty} \frac{e_{n+1}}{e_n^2} = M < \infty$$

for some M. Note that this is always a discrete notion (we have $n \in \mathcal{N}$ and $n \to \infty$; there is no equivalent with $h \in \mathbb{R}$ and $h \to 0$).

If we have $x_n \to x$ for some $x \neq 0$, then we can still do the same test, but apply it to the sequence $e_n \stackrel{\text{def}}{=} x_n - x$.

18. Does $e_n = 1/n$ converge linearly to 0?

Solution:

No! This is tricky, because $\lim_{n\to\infty} e_{n+1}/e_n = 1$, meaning it converges, but we need it to converge to something strictly less than 1 in order to call it linear convergence.

19. What kind of plot makes $e_n = 1/n$ look like a straight line?

Solution:

Both axes should be logarithmic.

20. Describe how you would evaluate the polynomial $p(x) = 5x^3 + 2x^2 + 3x + 4$ using Horner's method.

Solution:

Using the order indicated by the parenthesis:

$$((5x+2)x+3)x+4$$

Chapter 2: root finding

1. What are pros/cons of the bisection method?

   Solution:

   Pros: it gives a rigorous estimate on the error (if implemented in exact arithmetic); it converges reasonably fast; it doesn't require you to know the derivative of the function.

   Cons: as we saw in the HW, it can be confused if you have numerical issue due to floating point, e.g., when f crosses the x axes where it shouldn't (due to roundoff errors); also, convergence is slower than Newton; it requires us to find an interval where f crosses the x axis, so it can't work for double roots (like $f(x) = x^2$, which touches but never crosses the x axis).

2. What are pros/cons of Newton's method?

   Solution:

   Pros: when it converges, it often converges VERY fast (for a simple root), and it still converges reasonably for double roots; it can work on functions with double roots, and in particular, you don't need to identify an interval over which f changes sign

   Cons: you need to program the derivative; it does slow down when the multiplicity of the root is bigger than 1; you need to start sufficiently close to the root, otherwise it can diverge.

3. True/False: all root-finding problems can be recast as fixed point problems?

   Solution:

   True. Write $f(x) = 0$ as $x = x - f(x)$.

4. True/False: for a root-finding problem, there is only one equivalent fixed point problem?

   Solution:

   False, there are many ways, such as $x = x - f(x)$ or $x = x - 2f(x)$.

5. When might the bisection method not work even when there is a root?

   Solution:

   When it is a double root, like $f(x) = x^2$

6. How might the bisection method give misleading results?

   Solution:

   As we saw in the HW, due to numerical issues, it can think there's a root in an interval when there really isn't.

7. What is the rate of convergence of the bisection method? What error is this? (e.g., is it an error about finding the fixed point? Is it an error in the objective f?)

   Solution:

   It converges linearly, with constant $1/2$, e.g., the error at iteration k is proportional to $(1/2)^k$. This is an error on the root itself, $|x_k - p|$ where p is the root and x_k is our guess (the midpoint). This is as opposed to a bound on $|f(x_k)|$ (we know this will go to 0 as $x_k \to p$ since we assume the function is continuous, but without bounds on f' we can't put one kind of error in terms of the other).

8. Does Newton's method always converge?

Solution:

No! If the function f is sufficiently smooth, and if we start our initial guess sufficiently close, then it will converge.

9. If Newton's method converges, is it always at a quadratic rate?

   Solution:

   No, we only can prove quadratic convergence if we have a simple root, otherwise it can converge more slowly (often linearly).

10. Are there any possible issues with modified Newton's method?

    Solution:

    Yes, there can be numerical issues as we saw in the HW, e.g., 0/0 (or almost 0 over almost 0) issues.

11. What controls the eventual convergence rate of fixed-point problem?

    Solution:

    If $x = g(x)$ is the fixed point problem, and if g is differentiable at the fixed point p, then $|g'(p)|$ controls the convergence; in particular, if $|g'(p)| = 0$ then we have super-linear (including possibly quadratic) convergence; if $|g'(p)| > 1$ then we'll probably diverge (we have to reason to believe we will converge); and if $|g'(p)| < 1$ but $\neq 0$ then we get linear convergence with rate $|g'(p)|$.

12. True/False: a *contraction* is a function f such that $\forall x, y$, $|f(x) - f(y)| < |x - y|$.

    Solution:

    False. We need $\exists L < 1$ such that $\forall x, y$, $|f(x) - f(y)| \leq L|x - y|$. In the equation asked in the problem, we could get arbitrarily close to 1, and that is bad. For example, $f(x) = x + e^{-x}$ is a contraction if we restrict to the interval $[0, b]$ for any $b < \infty$, but it is not a contraction on $[0, \infty)$ (to see this, note $f'(x) = 1 - e^{-x}$).

    This is kind of similar to the fact that if a_n converges, and $a_n \leq b$ then we know $\lim_{n\to\infty} a_n \leq b$. However, if we know $a_n < b$, we *cannot* conclude $\lim_{n\to\infty} a_n < b$ (just think of $a_n = -1/n \to 0$).

13. If f is a linear function, what do we know about Newton's method?

    Solution:

    It will converge in 1 step.

Chapter 3: interpolation

1. Polynomial interpolation is a good idea if we have many noisy data points and want to approximate them with something smooth

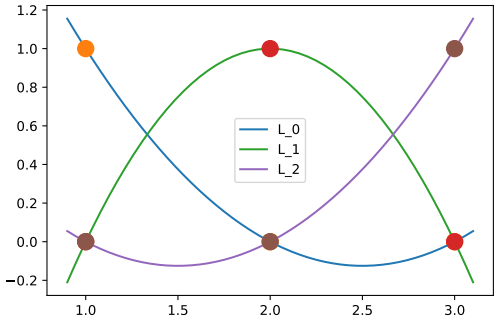
   Solution:

   No. Polynomial interpolation is not good if you have many points; do a piecewise interpolation instead, i.e., a spline. High-degree interpolation is particularly bad if you have noisy data.

2. For nodes $\{x_0 = 1, x_1 = 2, x_2 = 3\}$, draw a rough sketch of the Lagrange polynomials

Solution:

$L_2, 0$ interpolates 1 at x_0 and 0 at x_1 and x_2; similarly for the others. We know they are quadratics, and can guess their shape roughly. See figure for exact shape.



3. Describe the idea behind Lagrange interpolation

   Solution:

   We have Lagrange polynomials that interpolate the values 1 and 0 on the nodes, and these are easy to construct by inspection, and then it's easy to write the polynomial as a linear combination of these Lagrange polynomials because the coefficients are just the function values.

4. Find the minimum degree interpolating polynomial for nodes $\{-1, 0, 1\}$ and values $\{2, 3, 4\}$.

   Solution:

   For a small problem like this, the Lagrange formula is probably easiest. We have $L_{2,0}(x) = \frac{(x-0)(x-1)}{(-1-0)(-1-1)} = \frac{x(x-1)}{2}$, and $L_{2,1}(x) = \frac{(x--1)(x-1)}{(0--1)(0-1)} = \frac{x^2-1}{-1}$ and $L_{2,2}(x) = \frac{(x--1)(x-0)}{(1--1)(1-0)} = \frac{x(x+1)}{2}$. Then the actual interpolant is

   $$p(x) = 2L_{2,0}(x) + 3L_{2,1}(x) + 4L_{2,2}(x) = x(x-1) - 3(x^2-1) + 2x(x+1) = x + 3.$$

   So while usually this is a degree 2 polynomial, in this case it happens to be a degree 1 polynomial.

5. Describe the idea behind Newton's divided difference formula

   Solution:

   To find the coefficients of the Newton polynomials $1, (x-x_0), (x-x_0)(x-x_1), \ldots, \prod_{i=1}^{n-1}(x-x_i)$. We can find the coefficients using the recursively defined "divided differences." [Note: a question this vague will not be on the exam!]

6. How is $f[x_4, x_5, x_6]$ defined?

   Solution:

   as $\frac{f[x_5,x_6]-f[x_4,x_5]}{x_6-x_4}$.

7. What are pros/cons of Lagrange vs Newton interpolation?

   Solution:

   Lagrange, if you don't do the barycentric formula, can be unstable and is slow, taking $O(n^2)$ to evaluate a new point, and cannot be efficiently updated if you add a new point. With the barycentric formula, it takes $O(n^2)$ to precompute but then just $O(n)$ to evaluate, and it does have a fast update. The precomputation step depends only on nodes, not the values.

   Newton's divided differences takes $O(n^2)$ to compute the polynomial (which does depends on the values), but then only $O(n)$ to evaluate it at any point, so it's faster than the naive Lagrange method,

and similar speed to Lagrange with the barycentric formula. One disadvantage of the Newton divided difference method is that the precomputation depends on the values, not just the nodes, so if you want to interpolate many functions at the same set of nodes, you can't reuse the precomputation.

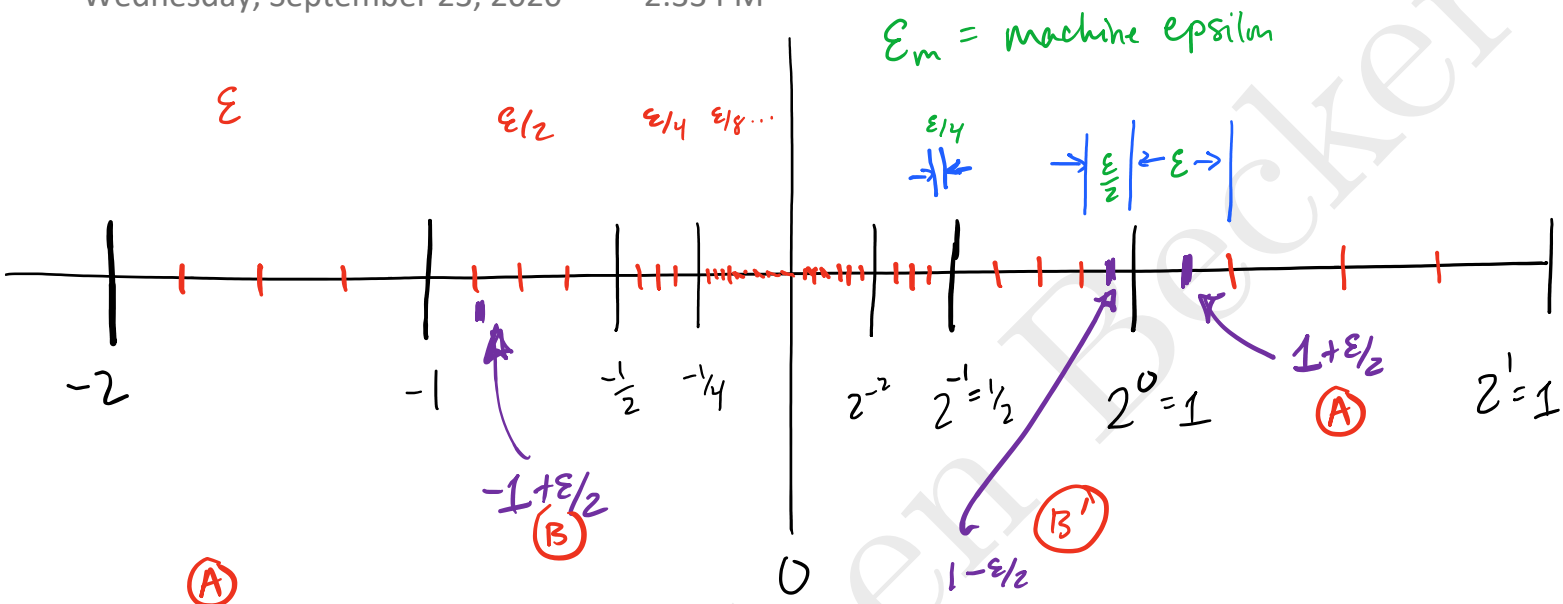
8. What is the Runge phenomenon. What are two ways to avoid it?

Solution:

It's the phenomenon that with many equispaced points, the interpolating polynomial starts to wiggle a lot near the end points, so if we're trying to approximate a smooth function f, the polynomial may be very far away from it. Two ways to avoid it: (1) do piecewise approximation, e.g., Splines; (2) space the points to have more density near the boundaries, in particular following the Chebyshev nodes.

Extra Floating Point diagram

Wednesday, September 23, 2020 2:33 PM



(A) $(1+\varepsilon/2)-1 = 1-1 = 0$

$= 1$ in floating pt.

(B) $1+(\varepsilon/2-1) = 1-(1-\varepsilon/2) = \varepsilon/2$ (B')

this is representable in floating pt.

Two definitions of ε_{mach}

(1) $1+\varepsilon_{mach}$ is next floating pt. number after 1

(what I drew in picture)

This is "variant" in wikipedia

$\varepsilon = 2^{-52} \approx 2.22e\text{-}16$

(2) Smallest number such that $1+\varepsilon_m \neq 1$

ie., "unit roundoff"

$\varepsilon = 2^{-53} \approx 1.11e\text{-}16$

if we "round to nearest" then

this is $1/2$ the value from definition (1)

In practice, we care that $\varepsilon_{mach} \approx 10^{-16}$,

specific numbers not always important